NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

DETECTING A MULTI-HOMED DEVICE USING CLOCK SKEW

by

Bryan J. Martin

September 2016

Co-Advisor:	Murali Tummala
Co-Advisor:	John C. McEachen

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704–0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.

1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** September 2016	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE DETECTING A MULTI-HOMED DEVICE USING CLOCK SKEW	**5. FUNDING NUMBERS**
6. AUTHOR(S) Bryan J. Martin	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB number ____N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.	**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

The aim of this thesis was to determine the feasibility of identifying a device connected to the Internet through multiple interfaces (i.e., multi-homed) using only the information provided by passively observing network traffic. Since multi-homed hosts allow an alternate means for outside entities to circumvent the security of a firewall and gain access to a network, it is important for a network's security to be able to detect and remove such devices. In this work, the idea of using clock skew—which is the difference in perceived time between two system clocks—as a unique signature is utilized to identify hosts on a network that are potentially multi-homed. Testing was done on a software-defined network that contained a multi-homed host. After traffic between hosts was collected and analyzed, analysis of the confidence intervals of the device's clock skew was conducted to determine if IP addresses originating from the same host could be successfully detected solely from network traffic. Testing confirmed that the proposed scheme provided a valid means of detecting a multi-homed device on a network. This scheme was repeated on multiple hosts and on a device with multiple connections to the network.

14. SUBJECT TERMS software defined network, multi-homed host, network monitoring, network fingerprinting	**15. NUMBER OF PAGES** 61
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540–01-280-5500

Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

DETECTING A MULTI-HOMED DEVICE USING CLOCK SKEW

Bryan J. Martin
Lieutenant, United States Navy
B.S., United States Naval Academy, 2008

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 2016

Approved by: Murali Tummala
Co-Advisor

John C. McEachen
Co-Advisor

R. Clark Robertson
Chair, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The aim of this thesis was to determine the feasibility of identifying a device connected to the Internet through multiple interfaces (i.e., multi-homed) using only the information provided by passively observing network traffic. Since multi-homed hosts allow an alternate means for outside entities to circumvent the security of a firewall and gain access to a network, it is important for a network's security to be able to detect and remove such devices. In this work, the idea of using clock skew—which is the difference in perceived time between two system clocks—as a unique signature is utilized to identify hosts on a network that are potentially multi-homed. Testing was done on a software-defined network that contained a multi-homed host. After traffic between hosts was collected and analyzed, analysis of the confidence intervals of the device's clock skew was conducted to determine if IP addresses originating from the same host could be successfully detected solely from network traffic. Testing confirmed that the proposed scheme provided a valid means of detecting a multi-homed device on a network. This scheme was repeated on multiple hosts and on a device with multiple connections to the network.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

IP	Internet protocol
ISP	Internet service provider
MAC	media access control
NAT	network address translation
NIC	network interface card
RTO	retransmission timeout
RTT	round-trip time
SDN	software-defined network
SSH	secure shell
TCP	transmission control protocol
TSecr	timestamp echo response
TSval	timestamp value

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

Thank you to my professors and advisors who guided me through the process of completing this thesis. And to my friends and family, thank you for your support. Hope kindly delivered us to our end goal.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Network security remains a major concern for all communications systems. With the advent of panoptic, or comprehensive, network management techniques such as software-defined networking (SDN), the ability of a system administrator to leverage the monitoring functions of a panoptic controller have led to the development of a large range of applications for network control and security to include monitoring applications for maintaining the security and integrity of one's network [1].

A. BACKGROUND AND MOTIVATION

A variety of security and cyber related concerns exist for any network. Before an attack can be conducted on a network, an attacker must first gain access. One method to prevent this is the use of a firewall between a private network and the Internet. A potential security flaw in a network is the existence of a multi-homed host [2]-[4]. Through the use of multiple interfaces on a host, the security of a network and the integrity of its firewall can be circumvented.

A multi-homed host is a device connected to the Internet through multiple interfaces [2]-[4]. If one of these connections is to a private network and the other to the open Internet, this provides a possible access vector that bypasses the network's firewall [4]. This threat calls for the need to be able to detect if a multi-homed host exists on a network and is the motivation behind this research.

B. THESIS OBJECTIVES AND APPROACH

The goal of this thesis is to develop a scheme for detecting multi-homed hosts in a panoptic network such as a SDN. A framework for an application that can be used to detect hosts using multiple interfaces that are independent of their Internet Protocol (IP) or Media Access Control (MAC) address is provided in this thesis.

The objective of this thesis is to identify techniques and monitoring schemes that can be used to increase the security of a network. In this thesis, we investigate the use of the clock skew of a host compared to a designated fingerprinter as a unique identifier. If a unique clock skew correlates to two or more unique IP addresses on the network, this represents a possible multi-homed device. In this work, analyses are conducted based on the confidence intervals of the calculated clock skews to determine if two similar clock skews represent the same, multi-homed host.

C. RELATED WORK

The idea for using the clock skew of a host for remote physical device fingerprinting was first suggested in [5]. It was shown that modern computer chips had detectable and distinguishable clock skews that could be calculated by observing the Transmission Control Protocol (TCP) timestamps from traffic on the network. It was then verified that the clock skew of a device remained constant even when using separate Ethernet and Wi-Fi interfaces originating from the same device [6].

This idea was further used as an enumeration tool in [7]. Researchers used clock skews of a device to determine the number of hosts active behind a network address terminal (NAT). This was accomplished by counting the number of unique clocks skews encountered from traffic exiting a NAT and correlating them to unique devices [7].

In this thesis, these ideas are expanded upon, and they are used to detect multi-homed devices active on a SDN. Since the clock skew of a device is constant and independent of the interface used, it can be used as a fingerprint for a multi-homed device. We also conduct the confidence interval analysis of the clock skew data encountered on the network to identify devices that appear to be separate based on IP address but are originating from the same device.

D. THESIS ORGANIZATION

The remainder of this thesis is organized as follows. In Chapter II, the security threats posed by a multi-homed host, the architecture and routing procedures within a SDN, and the system clock and its unique properties are introduced. The proposed scheme for multi-homed device detection is described in detail in Chapter III, while the results of the experiment are contained in Chapter IV. A description of the network that was used to test the feasibility of using clock skews to detect the presence of a multi-homed host is included. Finally, the thesis is concluded in Chapter V, where significant results and recommendations for future work are presented.

II. BACKGROUND

Network security continues to be a vital concern for a constantly connected society. One such concern is the access afforded to a network via a multi-homed host [2]. Mitigating the threat on a SDN by detecting such a device is the focus of this research. Before proposing the detection scheme for such a device, the relevant background information is presented in this chapter to introduce the threats and tools that are used to mitigate them. First, the basics of a multi-homed host and how such a device can be used to bypass a network's security are discussed. Then, the architecture and routing procedures of a SDN are presented. The system clock of a network device is discussed, and how it can to be used as a unique identifier is presented. Lastly, the concepts of confidence intervals and their role in hypothesis testing are described.

A. MULTI-HOMED HOST

A multi-homed host is one that has multiple connections to a network or networks. This can be accomplished by having multiple network interface cards (NICs) installed in the same host, which provides a host with multiple MAC and IP addresses [3]. Multi-homed hosts are used in a network for redundancy purposes [2]. With a multi-homed host on a network, the reliability of a network's access can be increased. Access node failure can be mitigated, and the connectivity from an Internet service provider (ISP) can be made more reliable by having separate connections to separate ISPs [8].

1. Security Threats with Multi-Homed Hosts

The threat from a multi-homed host comes from the fact that a multi-homed host can be used to bypass the firewall between an internal network and the Internet [2]. Certain operating systems, such as Windows, were never meant to isolate two interfaces within a host and often integrate traffic from one to the other [2]. This results in the ability for an infection on one network to be passed to another.

Closed networks are protected from the Internet by firewalls, which only allow designated traffic to flow between the two mediums. If a host is multi-homed, this allows for the opportunity to bypass the firewall and provide access to a closed network [4]. Once access to a host on a closed network is gained, potential threats can map a network and begin an exploitation process or infect the network with malicious code. An example of such a network configuration is depicted in Figure 1. Throughout this research, the threat of a multi-homed host serving as an access vector to a network is to be mitigated by the ability to detect the presence of such a host on the network.

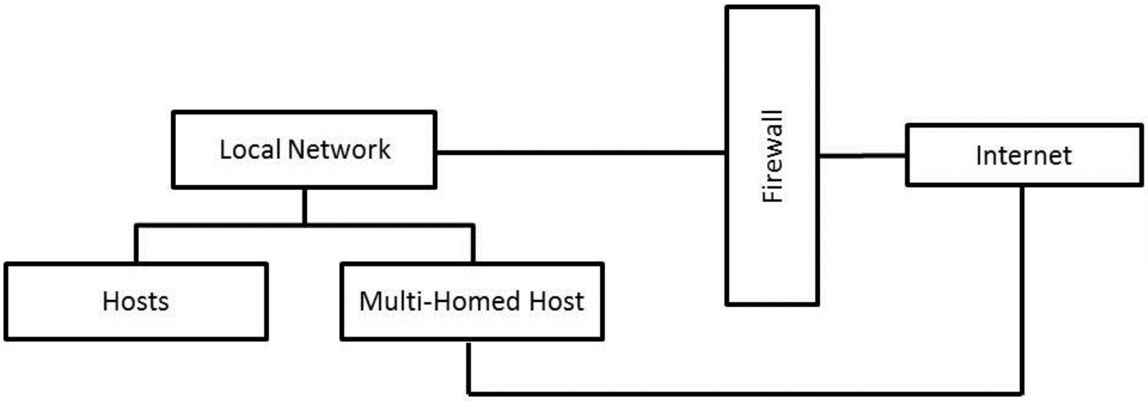


Figure 1. A Network with a Multi-Homed Host Implemented to Bypass the Firewall to the Internet

B. SOFTWARE-DEFINED NETWORKS

A software-defined network is an innovative networking scheme in which the control and data planes within a network are logically separated. In a SDN, the routing functions for the network are controlled from a centralized location, known as the controller [1], [9], [10]. This centralized controller is able to view the operation of the entire network, allowing it to monitor and react to any potential hazards that may exist [10].

1. Architecture

A SDN is divided into three planes that each interact to control the functionality of the network as shown in Figure 2. The lowest plane is the data plane, which consists of switches that forward packets based on flow rules [1]. Above the data plane is the control plane. From here, network traffic is monitored, and the flow rules for designated packets are determined [1], [9], [11]. The controller can be programmed by applications, allowing the network to dynamically react to any changes within the network. This is done at the upper plane, known as the application plane of the network [1], [11].

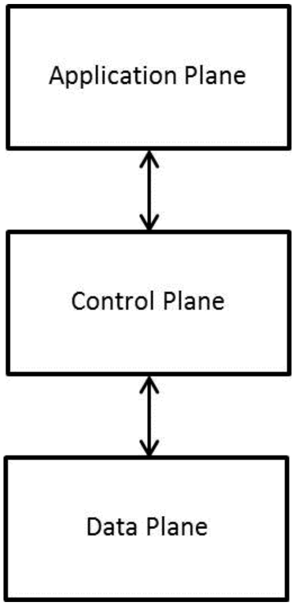


Figure 2. Functional Planes within a Software-Defined Network. Source: [1].

2. Routing

Routing within a SDN is completed using flow rules that are determined at the control plane and stored at the data plane. Due to this functionality, routing is now a rule-based process vice a destination-based process [1]. A SDN operates as a Transmission Control Protocol/Internet Protocol (TCP/IP) network and uses the OpenFlow protocol for its rule-based routing. The OpenFlow protocol matches packets to designated flow rules within a flow table at the data plane. If no such rule exists, the packet is forwarded to the control plane where a decision is made as to how it should be routed. Once this determination is made, the packet is forwarded back to the data plane for routing along with updates for the flow tables for future routing decisions [10], [11].

C. SYSTEM CLOCKS

Networked devices all have internal electric clocks that are built from both hardware and software components. These clocks control all timing functions for the device [12]. Within these electronic clocks, crystal oscillators are used to determine the clock signal and the rate at which the clock ticks [13]. These crystal oscillators each operate at different, unique frequencies due to the crystal type, the manufacturing parameters, and the small imperfections that are inherent to all manufacturing procedures [13], [14]. Due to these factors, clocks within a device operate at slightly different frequencies independent of clock type or manufacturing series [13]. This makes the system clock within a device a unique characteristic that can be exploited to identify that device.

1. TCP Timestamps

The TCP header consists of a standard 20 bytes of information followed by a portion of data allocated to options within the protocol [15], which is shown in Figure 3. In the options section of the header of a TCP packet is a field for the TCP timestamp. The TCP timestamp is a one-up counter based off a device's system clock that was introduced in RFC 1323 as a means of accurately measuring the round-trip time (RTT) between two devices. The need for accurately measuring the RTT of a packet is to provide a basis for determining the retransmission timeout interval (RTO) for lost or unacknowledged packets [16].

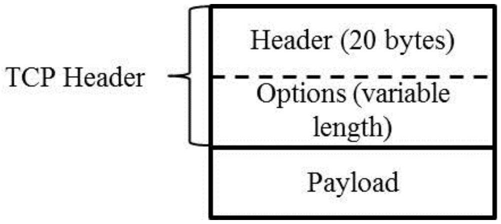


Figure 3. TCP Header with the Options Segment. Source: [15].

The TCP timestamp value is determined by a virtual "timestamp clock" that is based on the frequency of operation of the device's system clock. By observing the values of TCP timestamps, one can observe the operation of the system clock [5]. The TCP timestamp is a second-order effect of the system clock and is the means in which the clock skew is calculated in this research.

2. Clock Skew

The clock skew of a device is the difference in the operating frequencies of its system clock relative to the clock frequency of another device [5]. It is this parameter that can be used to identify the device based solely through passively observing network traffic. When using clock skew as a unique identifier, the identifier is valid only in relation to a designated device. This device is known as the fingerprinter [5]. In a SDN, the controller can be designated as the fingerprinter due to its ability to monitor all devices connected to the network.

D. CONFIDENCE INTERVALS

Confidence intervals are used in this research to bound the uncertainty of the calculated clock skews due to the randomness of the data collected and because the true mean value of the clock skew μ cannot be exactly measured or known. The clock skew is a random variable α that is assumed to be Gaussian with a density function $f(\alpha)$. A confidence interval provides a range of values in which the true calculated mean value lies with a specified probability 1-ε [17].

The confidence interval is defined as the range of C_L to C_U such that

$$P[C_L \le z \le C_U] = C, \tag{1}$$

where C is the desired confidence probability between zero and one for a given parameter z [17]. The value C=1-ε, where ε is the acceptable error. The bounds of a confidence interval for the density function $f(\alpha)$ with an accepted error ϵ and true mean μ are shown in Figure 4. The bounds of this confidence interval C_L and C_U are determined by solving [18]

$$\frac{\varepsilon}{2} = \int_{C_U}^{\infty} f_\alpha(\alpha)d\alpha \tag{2}$$

and

$$\frac{\varepsilon}{2} = \int_{-\infty}^{C_L} f_\alpha(\alpha) d\alpha \ . \tag{3}$$

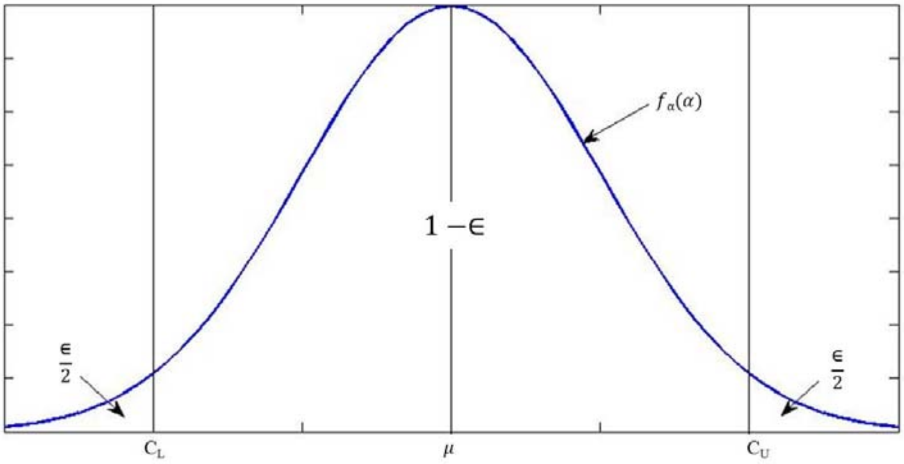


Figure 4. The Bounds C_L and C_U of the Confidence Intervals of a Given Density Function with a True Mean μ and an Acceptable Error ϵ. Source: [18].

Confidence intervals are used in hypothesis testing to decide between two possible scenarios. If a hypothesis H_0 is made about a parameter and that parameter falls within the range of a confidence interval, then that hypothesis is accepted with a confidence level of C [17]. This idea is used in this thesis to analyze the clock skews of the devices on the network to determine if they originate from the same device.

The possible threats to a network from devices known as multi-homed hosts, devices with more than one interface connected to a network was introduced in this chapter. We then described the functionality of a SDN and the advantages of such a network as compared to traditional routing procedures. Finally, the system clock of a device was described in detail along with how to measure its unique operating parameters and use that information as a unique identifier for the device. This information is utilized in Chapter III and demonstrated in Chapter IV in a scheme for detecting a multi-homed device active on a SDN.

III. MULTI-HOMED DEVICE DETECTION SCHEME USING CLOCK SKEW

The problem we present in this research how is to detect a host on a network using multiple connections through multiple NICs. In order for a solution to be achieved, we must determine whether or not the traffic between different IP addresses can be correlated in order to determine if those IP addresses belong to a multi-homed host. Two assumptions are made in developing the proposed scheme. The first is that passive means of collection are used over the network. The second is that the observer can observe and collect traffic from all IP addresses of a multi-homed host.

The rest of this chapter is organized as follows. First, the proposed scheme for detecting multi-homed hosts is presented based on a host's clock skew. Then, we discuss the network configuration and the method of generating and collecting traffic. Finally, TCP timestamps are described, and the method of calculating the clock skew of a host is presented.

A. PROPOSED SCHEME

The proposed solution is to collect TCP timestamp data from a host in order to calculate its clock skew for use as a fingerprint. The clock skew of a host is unique and has very little variation over time. It has been demonstrated that the clock skew of a host stays relatively constant even if two interfaces (Ethernet and Wi-Fi) are used to connect to a network. For these reasons, the clock skew can be used as an identifier for a given host [5], [6], [19]. The aim of this thesis is to determine whether multiple IP addresses with similarly calculated clock skews are from the same device.

The first step in the proposed process is to monitor and collect traffic across the network. The traffic of interest is the TCP segments exchanged between hosts, specifically those containing TCP timestamps. From this information, the clock skew of each host relative to a central host (the fingerprinter) can be calculated. After the clock skews of each host on the network are determined, analysis is conducted based on hypothesis testing using confidence intervals to identify potential multi-homed hosts. A testbed network with a fingerprinter is shown in Figure 5, and the process for detecting a multi-homed host is outlined in Figure 6.

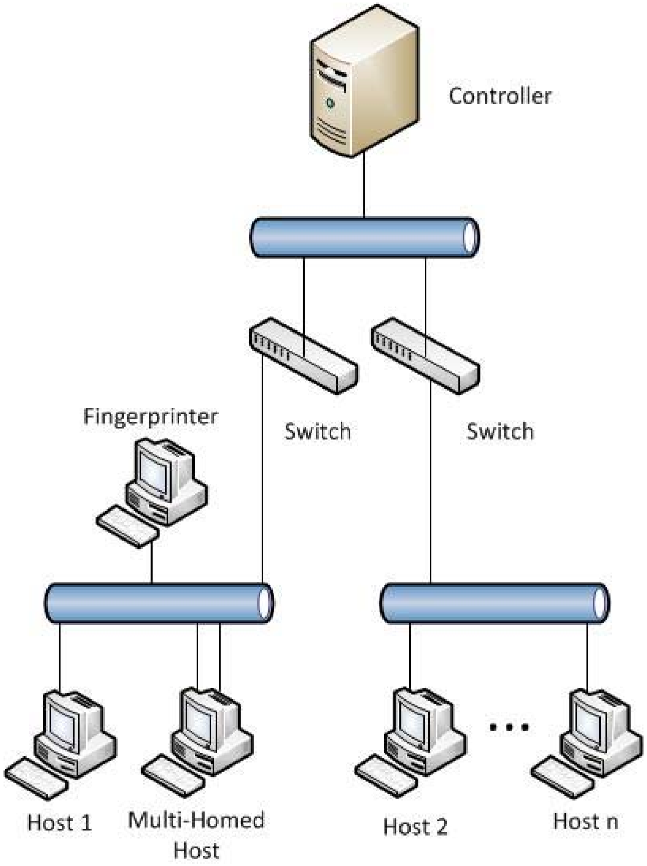


Figure 5. Generic Network Configuration of a SDN with a Controller, Two Switches, and n Number of Hosts with One Acting as the Fingerprinter for Testing and Another Multi-Homed

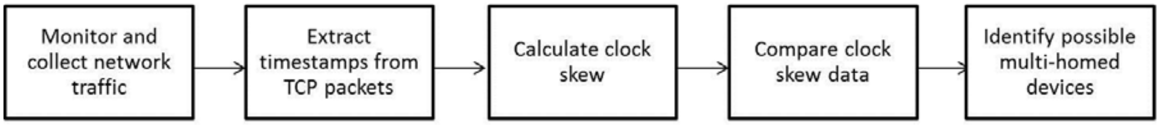

Monitor and collect network traffic
Extract timestamps from TCP packets
Calculate clock skew
Compare clock skew data
Identify possible multi-homed devices

Figure 6. Process of Detecting Multi-Homed Devices Using Clock Skew

In previous work, this method was utilized for the determination of the number of hosts behind a NAT. It was suggested in [5] and shown in [7] that one could determine the number of hosts sending traffic through a NAT by calculating and comparing the unique clock skews encountered. In this thesis, we propose the use of correlating clock skews between multiple IP addresses to determine if they are originating from the same, multi-homed device.

B. NETWORK CONFIGURATION

To test our proposed scheme, we collect and analyze traffic from hosts on a network. A version of the network layout is shown in Figure 7. Multiple hosts are connected to each switch with one host among them being multi-homed. The multi-homed host uses separate Ethernet connections to connect to the network. A central host acts as the fingerprinter for determining the clock skews of all hosts on the network [5]. The fingerprinter is chosen so that it has the ability to observe traffic from both connections of the multi-homed host.

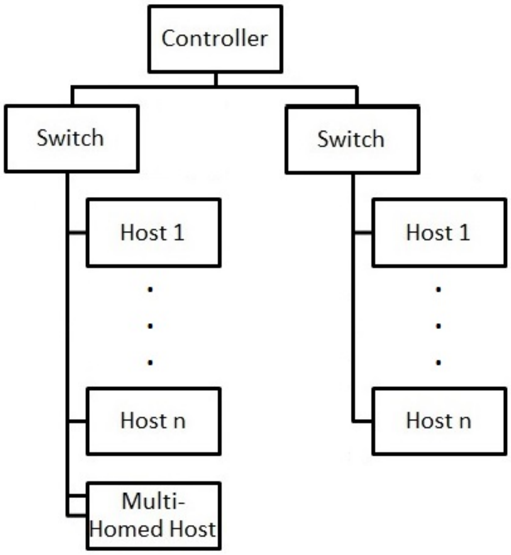


Figure 7. Configuration of a SDN with a Multi-Homed Host Connected to a Single Switch

C. CLOCK SKEW

In order to test the proposed scheme, network traffic containing TCP segments with timestamps was collected. Using this data, the clock skew of each host can be calculated.

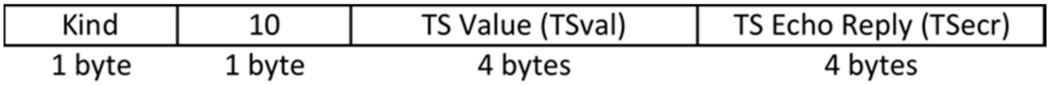
1. Traffic Monitoring

The fingerprinter monitors the network from a centralized location for TCP segments in the network traffic. Not all of these TCP segments seen in the network traffic will contain timestamps. It is the segments with TCP timestamps originating from a host and sent to the fingerprinter that are of interest. These segments are aggregated, and the TCP timestamps are used in the calculation of the clock skew. These TCP timestamps are collected along with the time of collection based on the fingerprinter's own clock. The calculation for the clock skew based on this data is discussed in more detail later in this chapter.

2. TCP Timestamps

TCP timestamps were introduced as a means to provide a simple and accurate tool to measure the RTT of a packet transmission [16]. TCP is meant to be a reliable connection-oriented protocol, and this reliable connection is achieved by the retransmission of lost or dropped packets. The duration of time before retransmissions are sent is known as the RTO and is calculated by knowing the RTT of a packet. TCP timestamps provide a simple and accurate means of determining this RTT by sending and echoing relative timing information within the TCP packet [16].

The timestamp is included in the TCP options portion of the header and consists of 10 bytes of data. The format of these 10 bytes of data is shown in Figure 8. The first byte is the kind of timestamp, the second byte is the length of the option field, the next four bytes contain the current value of the sender's timestamp, and the final four bytes are an echo of the timestamp received [16].



Kind	10	TS Value (TSval)	TS Echo Reply (TSecr)
1 byte	1 byte	4 bytes	4 bytes

Figure 8. TCP Timestamp Options Field. Source: [16].

The value of the timestamp comes from a virtual internal clock that is known as the "timestamp clock" and is based upon the device's own clock [16]. TCP timestamps are a second-order effect of the host's system clock, and through their collection and measurement, the operation of a host's system clock can be observed [5].

3. Clock Skew

The clock skew is a physical trait of a host's processor caused by the different operating frequencies of crystal oscillators within electronic clocks. The discrepancy in operating frequencies is a product of the manufacturing process and results in small differences in clock speed of each clock [13], [19]. This difference in frequencies between the system clocks of separate devices is calculated as the first derivative of a function that includes the offset of their observed times [5], [6], [19].

Once the TCP timestamps have been collected, the clock skew can be calculated based on the procedure provided in [5]. The first step is to determine the time and TCP timestamp offsets of a collected packet versus the initial time of collection. The first packet collected by the fingerprinter from a host is used as the baseline for the offset. The time offset is given by [5]

$$x_i = t_i - t_1, \tag{4}$$

where x_i is the difference between the time of collection of the i^{th} packet at time t_i and the initial time of collection t_1. The timestamp offset w_i for the i^{th} packet is given by

$$w_i = \frac{T_i - T_1}{f}, \tag{5}$$

where T_i is the timestamp of the i^{th} packet, T_1 is the timestamp of the first packet at the initial time of collection and f is the operating frequency of the host's clock.

Once the time and timestamp offsets are known, the difference y_i between the observed time at the fingerprinter and the observed time from the source host based on its timestamps is calculated as

$$y_i = w_i - x_i. \tag{6}$$

Given the set of points x and y for the data collected, the set of offset values O_T for N collected packets is represented as

$$O_T = \{(x_i, y_i) : i \in \{1,...,\mathrm{N}\}\} \ , \tag{7}$$

and we model the data as a slope-intercept line equation.

The clock skew is the first derivative (or slope) α of this line

$$\alpha \cdot x_i + \beta \geq y_i \, , \tag{8}$$

with a *y*-intercept of β that fits the upper bound of the set of points O_T. The solution to (8) is obtained using a linear programming technique with the goal to minimize the objective function *J*

$$J = \frac{1}{N}\sum_{i=1}^{N}(\alpha \cdot x_i + \beta - y_i) \tag{9}$$

for *N* packets [5]. This procedure is repeated for each host on the network.

D. DETECTION OF MULTI-HOMED HOSTS

Once the clock skews have been calculated, a comparison must be made in order to determine which IP addresses represent the potential multi-homed host in the network. To improve accuracy, a large number of trials are required. Based on the central limit theorem, the sample mean of independent random variables approaches a Gaussian distribution [20]. Consequently, given a relatively large number of trials, we assume that the clock skews calculated for each host over these trials approaches a Gaussian distribution.

After the mean clock skew is determined for a host, analysis is done using the confidence intervals for the clock skew of all hosts and hypothesis testing to determine whether the IP addresses belong to a multi-homed host.

The sample mean $m_i$of the clock skew for the i^{th} host is determined as

$$m_i = \frac{1}{N}\sum_{i=1}^{N}\alpha_i \tag{10}$$

where α_{i} is the calculated clock skew for the i^{th} host [18]. Now we formulate the following hypothesis. The first hypothesis H_0 states that m_j for the j^{th} host's clock skew is within the range of the i^{th} host's confidence interval. The second hypothesis H_1 states that m_j for the j^{th} host's clock skew is outside of this range. The lower bound of the confidence interval for a host i is represented by $C_{L,i}$, and the upper bound is represented by the value $C_{U,i}$. If m_j falls within the confidence interval

$$C_{L,i} \le m_j \le C_{U,i}\,, \tag{11}$$

when $i \neq j$, then hypothesis H_0 is accepted and the IP addresses are flagged as originating from the same host. If not, then hypothesis H_1 is accepted and the IP addresses did not originate from the same host [20]. The process for this analysis is shown in Figure 9.

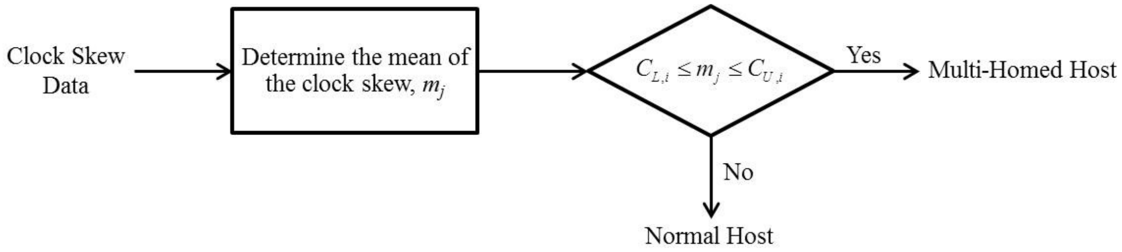

Clock Skew Data
Determine the mean of the clock skew, m_j
$C_{L,i} \le m_j \le C_{U,i}$
Yes
Multi-Homed Host
No
Normal Host

Figure 9. Process of Testing Hypotheses Using Confidence Intervals to Determine If Hosts Are Multi-Homed

In this chapter, a scheme for detecting a multi-homed host active on a SDN using information from TCP traffic on the network was introduced. From the observed TCP timestamps, the clock skew between an active device's system clock and the system clock of a designated fingerprinter can be calculated. This is a unique value for a device and can be used as an identifier for that device. The validity of the scheme is tested in Chapter IV using a SDN test bed.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. TESTING AND RESULTS

A scheme for calculating clock skew based on TCP traffic was proposed in Chapter III. This scheme was validated using a SDN test bed for data collection. The configuration of the network used and the means for generating and capturing the test traffic is described in this chapter. We then calculate the clock skew of each host and apply the confidence interval analysis on the clock skew of each host to identify the multi-homed host.

A. NETWORK CONFIGURATION

A portion of the SDN test bed that was built for testing in [21] was used in this experiment and consisted of two HP switches and seven Raspberry Pis as hosts. The switches used were the HP 2920 and the HP 3800, and the Raspberry Pis were connected to the network using their built-in 10/100 Mbps Ethernet connection. The network configuration that was used is shown in Figure 10.

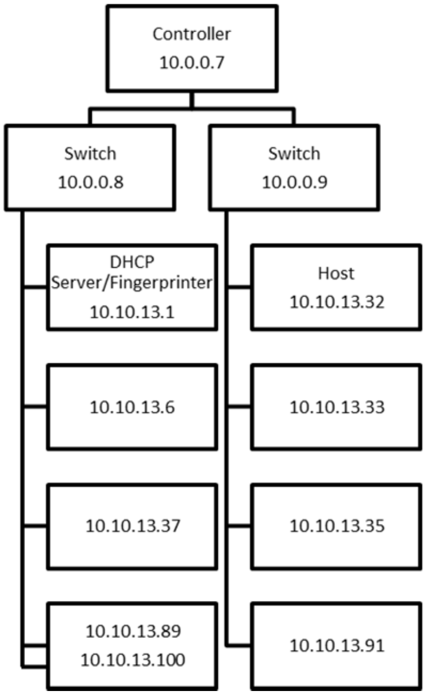


Figure 10. Network Configuration Used in Testing

One of the Raspberry Pis had an added USB 2.0 Gigabit LAN adapter that was used as its second connection to the network. The connections for this Raspberry Pi are shown in Figure 11. This was the dual-homed device used in testing and the host that was to be experimentally identified. This host used the IP addresses 10.10.13.89 and 10.10.13.100. Both connections from this host were connected to the HP 2920 switch.

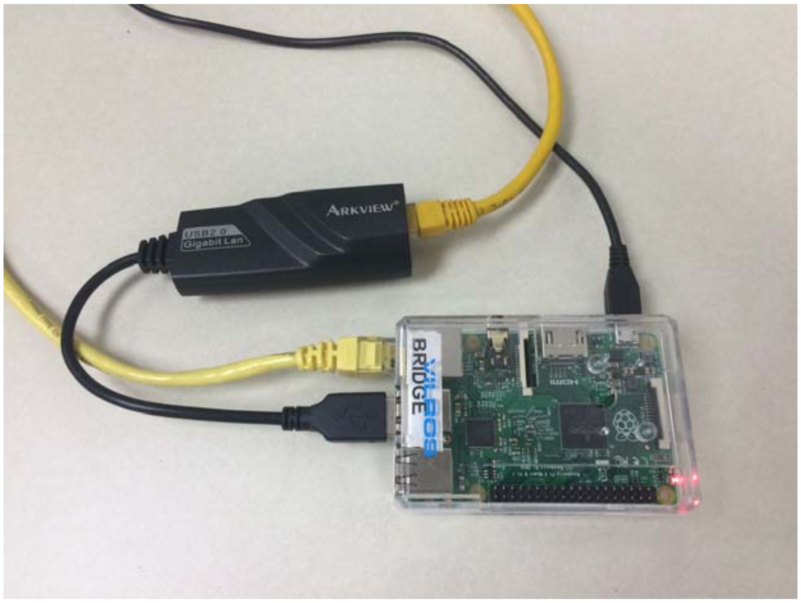
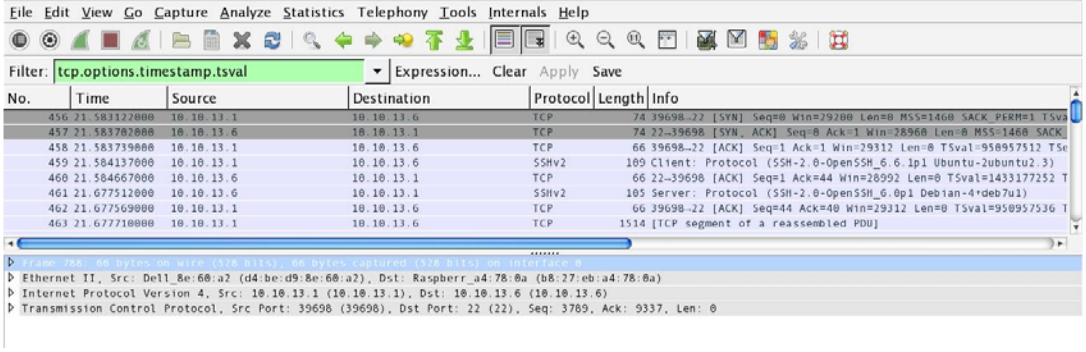


Figure 11. Dual-Homed Raspberry Pi Used in Testing

Also connected to the network was a Dell T1600 running Ubuntu that was acting as the DHCP server for the network. The DHCP server was used as the fingerprinter in this experiment and was chosen due to the fact that it maintained a static IP address of 10.10.13.1 throughout testing.

B. TRAFFIC GENERATION AND COLLECTION

In order to establish the necessary TCP connections for the purpose of creating TCP timestamps, traffic was generated by creating an Secure Shell (SSH) connection between the fingerprinter and the hosts on the network. This SSH connection allowed for the required TCP handshakes to be made and timestamps to be exchanged between the host and the fingerprinter for collection. Packets with TCP timestamps that were originating from a host were collected using Wireshark. An example from Wireshark of the initiation of the TCP connection is shown in Figure 12. The internals of the packet are depicted in Figure 13 with the TCP Options segment highlighted to show the timestamp value (TSVal) of the packet from 10.10.13.1 and the timestamp echo reply (TSecr) of the timestamp in the last packet from 10.10.13.6.

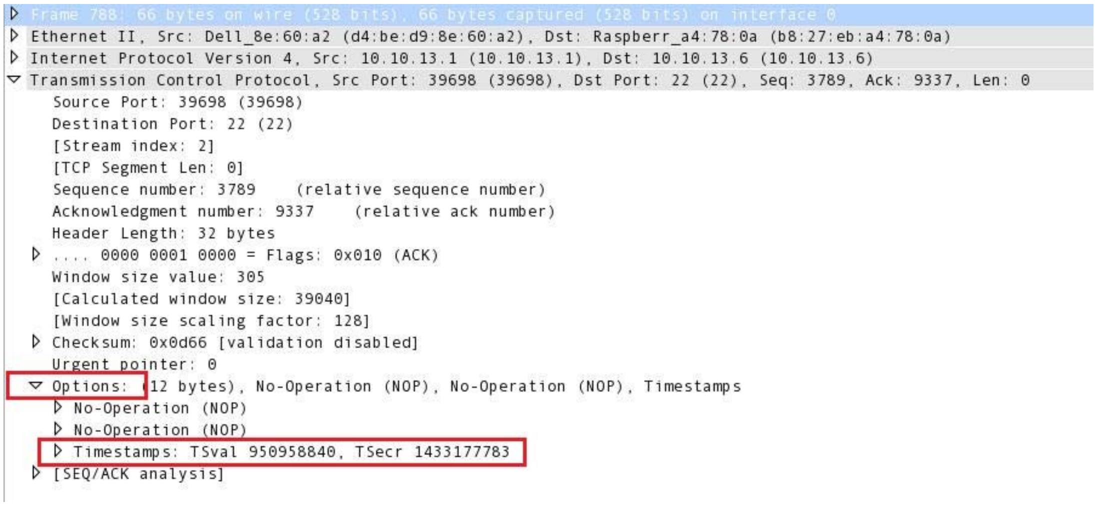

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply Save

Filter: tcp.options.timestamp.tsval

```
No.   Time            Source       Destination  Protocol Length Info
  456 21.583122000    10.10.13.1   10.10.13.6   TCP        74 39698→22 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSva
  457 21.583702000    10.10.13.6   10.10.13.1   TCP        74 22→39698 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_
  458 21.583739000    10.10.13.1   10.10.13.6   TCP        66 39698→22 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=950957512 TSe
  459 21.584137000    10.10.13.1   10.10.13.6   SSHv2     109 Client: Protocol (SSH-2.0-OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.3)
  460 21.584667000    10.10.13.6   10.10.13.1   TCP        66 22→39698 [ACK] Seq=1 Ack=44 Win=28992 Len=0 TSval=1433177252 T
  461 21.677512000    10.10.13.6   10.10.13.1   SSHv2     105 Server: Protocol (SSH-2.0-OpenSSH_6.0p1 Debian-4+deb7u1)
  462 21.677569000    10.10.13.1   10.10.13.6   TCP        66 39698→22 [ACK] Seq=44 Ack=40 Win=29312 Len=0 TSval=950957536 T
  463 21.677710000    10.10.13.1   10.10.13.6   TCP      1514 [TCP segment of a reassembled PDU]

▷ Frame 788: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
▷ Ethernet II, Src: Dell_8e:60:a2 (d4:be:d9:8e:60:a2), Dst: Raspberr_a4:78:0a (b8:27:eb:a4:78:0a)
▷ Internet Protocol Version 4, Src: 10.10.13.1 (10.10.13.1), Dst: 10.10.13.6 (10.10.13.6)
▷ Transmission Control Protocol, Src Port: 39698 (39698), Dst Port: 22 (22), Seq: 3789, Ack: 9337, Len: 0
```

Figure 12. TCP Connection Being Made Between Fingerprinter and Host



```
▷ Frame 788: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
▷ Ethernet II, Src: Dell_8e:60:a2 (d4:be:d9:8e:60:a2), Dst: Raspberr_a4:78:0a (b8:27:eb:a4:78:0a)
▷ Internet Protocol Version 4, Src: 10.10.13.1 (10.10.13.1), Dst: 10.10.13.6 (10.10.13.6)
▽ Transmission Control Protocol, Src Port: 39698 (39698), Dst Port: 22 (22), Seq: 3789, Ack: 9337, Len: 0
    Source Port: 39698 (39698)
    Destination Port: 22 (22)
    [Stream index: 2]
    [TCP Segment Len: 0]
    Sequence number: 3789    (relative sequence number)
    Acknowledgment number: 9337    (relative ack number)
    Header Length: 32 bytes
  ▷ .... 0000 0001 0000 = Flags: 0x010 (ACK)
    Window size value: 305
    [Calculated window size: 39040]
    [Window size scaling factor: 128]
  ▷ Checksum: 0x0d66 [validation disabled]
    Urgent pointer: 0
  ▽ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
    ▷ No-Operation (NOP)
    ▷ No-Operation (NOP)
    ▷ Timestamps: TSval 950958840, TSecr 1433177783
  ▷ [SEQ/ACK analysis]
```

Figure 13. TCP Portion of Packet Showing Timestamp Information

C. CLOCK SKEW CALCULATION AND RESULTS

Given the test traffic collected by Wireshark, the next step was to calculate the clock skew of each host. One hundred samples of data were collected at ten minute intervals, and MATLAB was used for calculations. Using the MATLAB function *linprog*, we solved (9) from Chapter III for each host. The solution provided the values of α and β, which are the slope and *y*-intercept of the solution to (8). The value of α corresponds to the clock skew and is the value of concern in this scenario.

The clock skew for each host was calculated independently for each trial using the MATLAB code in the Appendix. The upper-bound solution, which was used because the delays found within a network between hosts are all positive, for the set of points O_T was solved for each host [5]. As shown in Figure 14, the solution for the set of data points in red corresponding to host 10.10.13.100 provides a slope of 0.0000101203 or 10.1203 ppm for the line in blue representing the upper bound of the data set. This slope is the clock skew for this host when compared to the clock of the fingerprinter, 10.10.13.1.

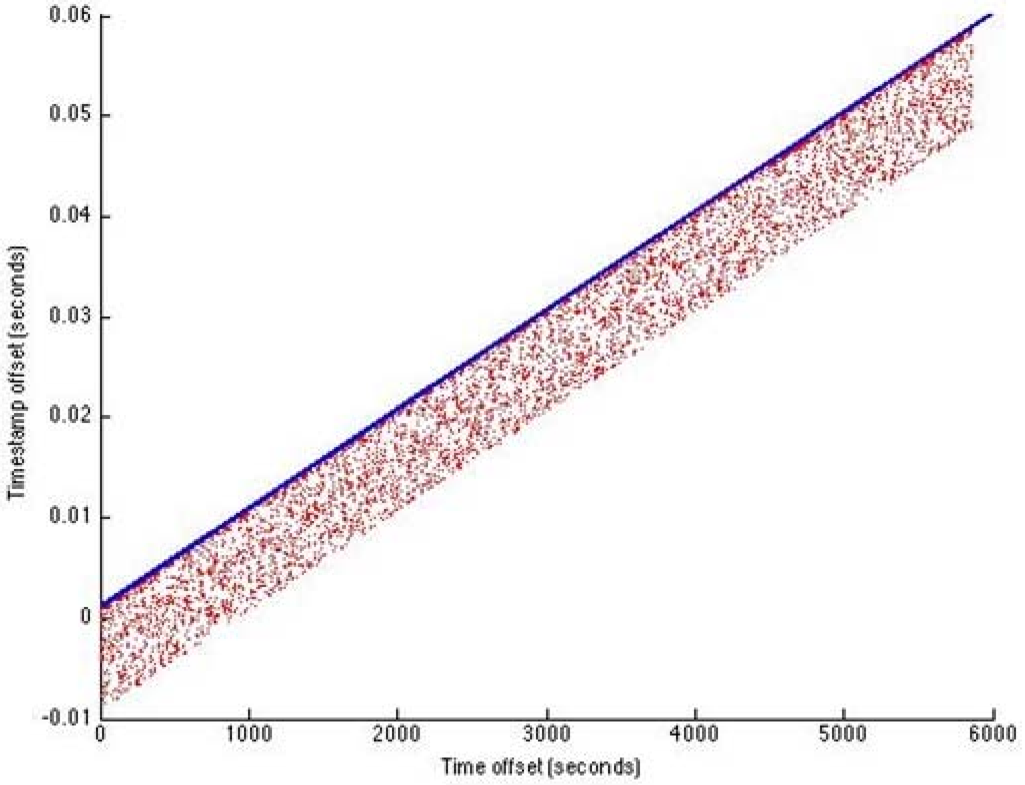


Figure 14. Upper-Bound Solution for Host 10.10.13.100 over a Single Trial

Comparing the slopes for the upper-bound solution of the data sets of all hosts over a single trial shows the variation of the clock skews found in this network. As seen in Figure 15, there is a range of positive and negative values for the clock skew corresponding to a host's clock being ahead of or behind the clock of the fingerprinter. The hosts using the IP addresses of 10.10.13.89 and 10.10.13.100 both have solutions with similar slopes and stand out as possibly being multi-homed due to the fact that the solution to (8) for each host appears to be represented by two parallel lines.

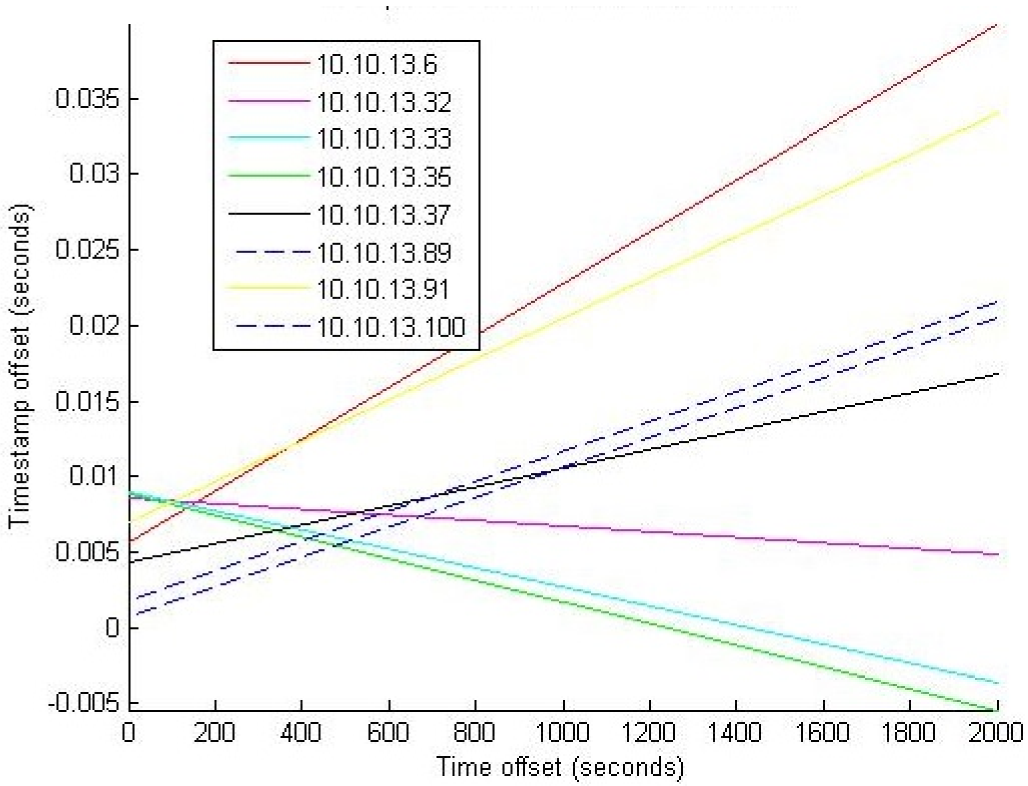


Figure 15. Upper Bound Solution of All Hosts over a Single Trial

The data in Figure 15 is supported by further trials. The mean value for each clock skew after 100 trials is depicted in Table 1. This data shows that the clock skews for 10.10.13.89 and 10.10.13.100 are similar. When compared to the differences between clock skews of the other hosts tested, as shown in Table 2, the difference between 10.10.13.89 and 10.10.13.100 appears to be negligible.

Table 1. Mean Clock Skew of All Hosts over 100 Trials (in ppm)

Host	Clock Skew (ppm)
10.10.13.6	17.126
10.10.13.32	-1.953
10.10.13.33.	-6.405
10.10.13.35	-7.313
10.10.13.37	6.700
10.10.13.89	10.132
10.10.13.91	13.020
10.10.13.100	10.140

Table 2. Difference of Clock Skew Between All Hosts (in ppm)

Host	10.10.13.6	10.10.13.32	10.10.13.33.	10.10.13.35	10.10.13.37	10.10.13.89	10.10.13.91	10.10.13.100
10.10.13.6	0.000	19.078	23.531	24.439	10.426	6.994	4.106	6.986
10.10.13.32	19.078	0.000	4.453	5.360	8.653	12.084	14.972	12.092
10.10.13.33.	23.531	4.453	0.000	0.908	13.105	16.537	19.425	16.545
10.10.13.35	24.439	5.360	0.908	0.000	14.013	17.445	20.332	17.452
10.10.13.37	10.426	8.653	13.105	14.013	0.000	3.432	6.320	3.440
10.10.13.89	6.994	12.084	16.537	17.445	3.432	0.000	2.888	0.008
10.10.13.91	4.106	14.972	19.425	20.332	6.320	2.888	0.000	2.880
10.10.13.100	6.986	12.092	16.545	17.452	3.440	0.008	2.880	0.000

For these comparisons and for the calculation of the confidence intervals, the data was assumed to approach a Gaussian distribution after the 100 trials. As shown in Figure 16, the range of clock skews collected for host 10.10.13.6 over these trials approaches a normal distribution.

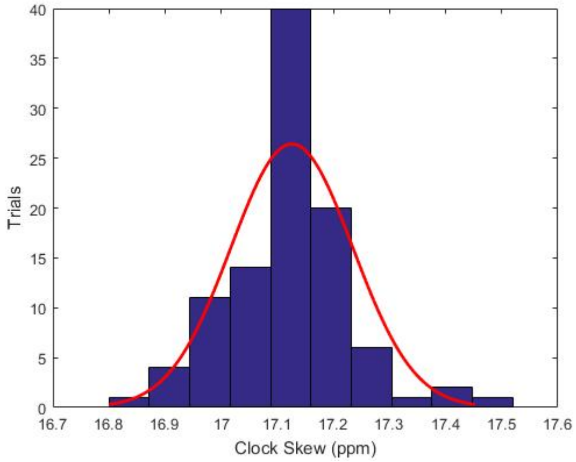


Figure 16. Histogram for the Calculated Clock Skews of Host 10.10.13.6 Over 100 Trials as they Approach a Gaussian Distribution

A 95% confidence interval for the clock skew of each host was calculated over the 100 trials conducted. The confidence interval was solved using the *paramci* function within MATLAB. The results for the confidence intervals are shown in Figure 17. In Figure 17 the value of the clock skew for each host is shown as a bar graph in blue. The error bar in red covers the range of values from the lower to the upper bounds of the confidence interval. The confidence interval for each clock skew is quite small, which suggests that the clock skew varies only slightly over time; this result has been observed in previous work [5], [6].

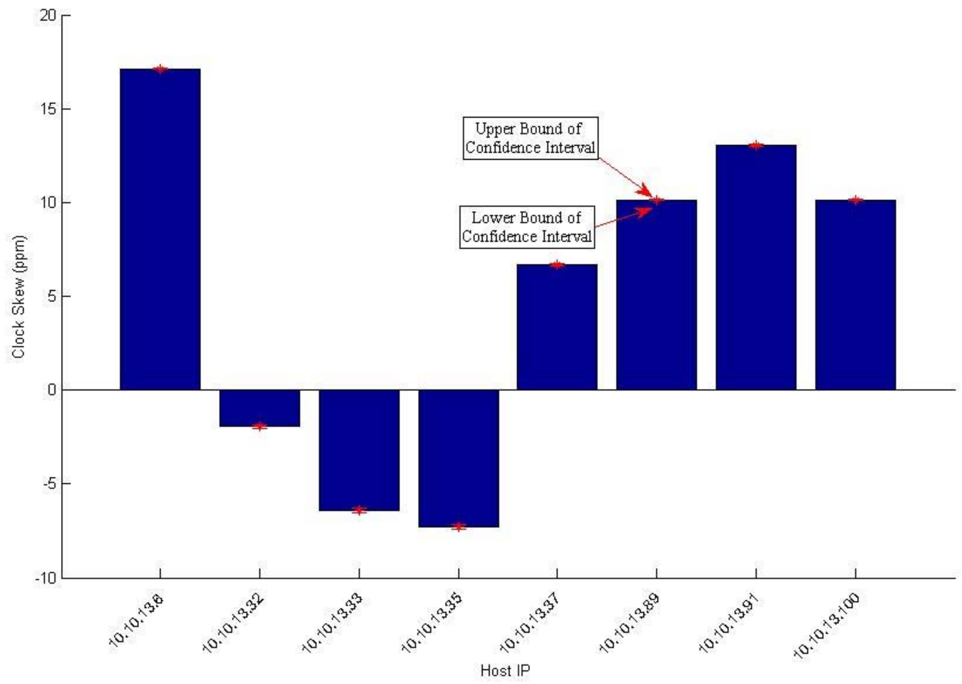
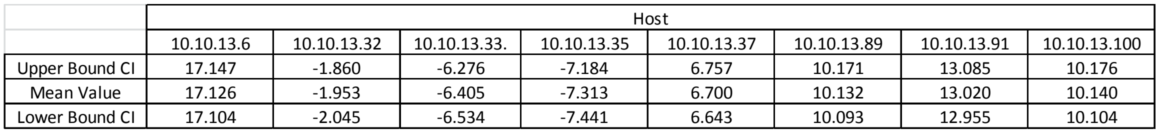


Figure 17. Confidence Interval of 95% for the Clock Skew of All Hosts over 100 Trials

D. DETECTION OF THE DUAL-HOMED HOST

As described in Chapter III, analysis of the confidence intervals of the clock skew for each host was used to determine which hosts were possibly multi-homed. Using the confidence intervals as presented in Figure 17, we applied the ideas presented in Chapter III to the given data. When the mean clock skew of each host is compared to the confidence interval calculated for all other hosts, the possible dual-homed host can be identified. The upper and lower bounds for the confidence interval for the clock skews of all hosts are shown in Table 3 along with the mean value of the clock skews calculated over 100 trials.

Table 3. Upper and Lower Bounds of the 95% Confidence Interval of Each Host's Clock Skew



	Host							
	10.10.13.6	10.10.13.32	10.10.13.33.	10.10.13.35	10.10.13.37	10.10.13.89	10.10.13.91	10.10.13.100
Upper Bound CI	17.147	-1.860	-6.276	-7.184	6.757	10.171	13.085	10.176
Mean Value	17.126	-1.953	-6.405	-7.313	6.700	10.132	13.020	10.140
Lower Bound CI	17.104	-2.045	-6.534	-7.441	6.643	10.093	12.955	10.104

When the mean value of each calculated clock skew is compared to the confidence interval of the clock skew for each host, it is observed that the possible dual-homed hosts are 10.10.13.89 and 10.10.13.100. The confidence intervals for all hosts are shown in Figures 18–25. The clock skews for the hosts 10.10.13.6, 10.10.13.32, 10.10.13.33, 10.10.13.35, 10.10.13.37 and 10.10.13.91 are shown in Figures 18–22 and Figure 24, respectively; the confidence intervals of the designated hosts are in blue while the values for clock skews for all hosts on the network in red. As can be seen in these figures, the confidence interval for a given host only includes the value of its own clock skew. In Figure 23 and Figure 25, the hosts represented by the IP addresses of 10.10.13.89 and 10.10.13.100 fall within each other's confidence interval, while the other hosts remain outside of these bounds. After comparing the data in Table 3 to Figure 23 and Figure 25, these results confirm the initial network setup where the hosts represented by the IP addresses 10.10.13.89 and 10.10.13.100 were from the same Raspberry Pi.

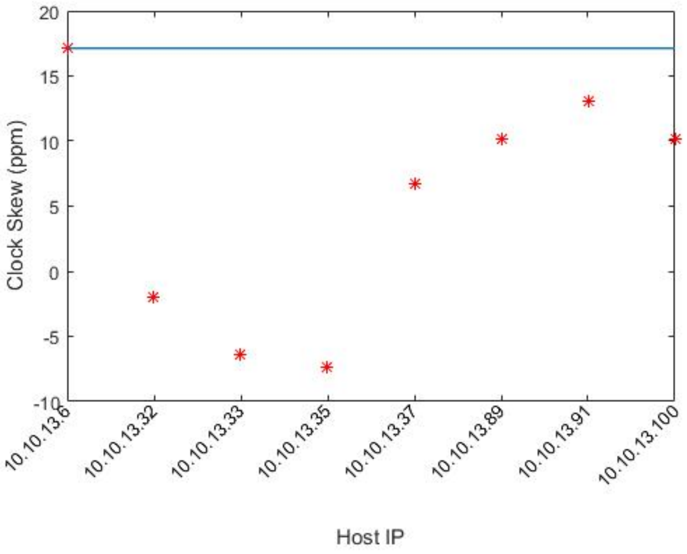


Figure 18. Confidence Interval of 10.10.13.6 Compared to the Mean Value of All Clock Skews Calculated

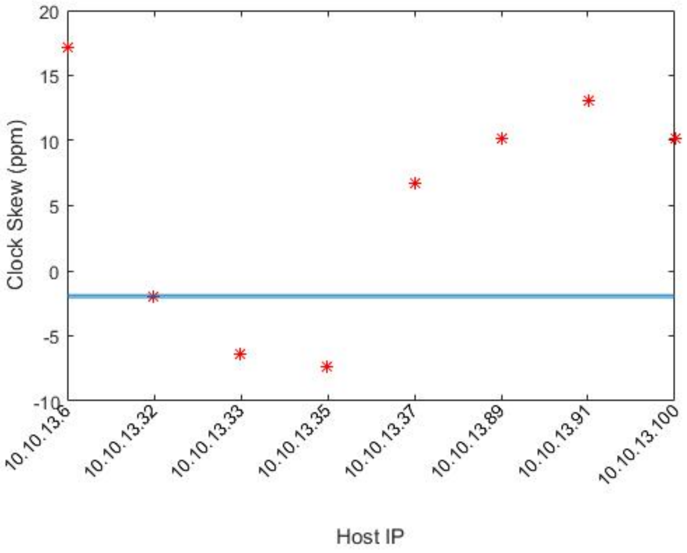


Figure 19. Confidence Interval of 10.10.13.32 Compared to the Mean Value of All Clock Skews Calculated

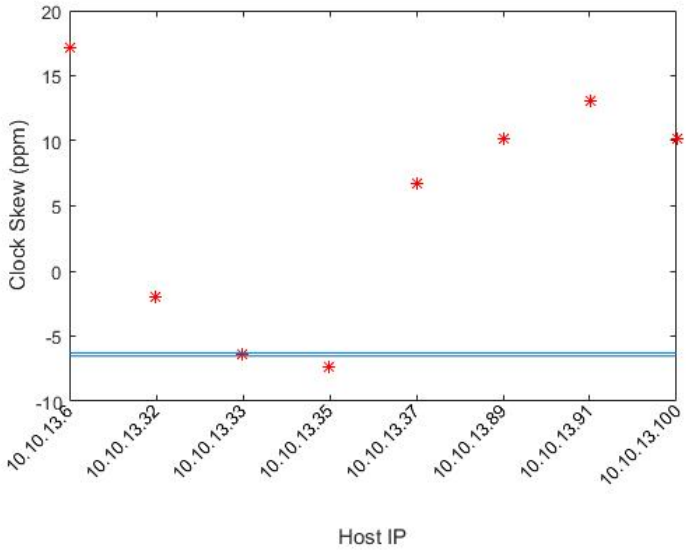


Figure 20. Confidence Interval of 10.10.13.33 Compared to the Mean Value of All Clock Skews Calculated

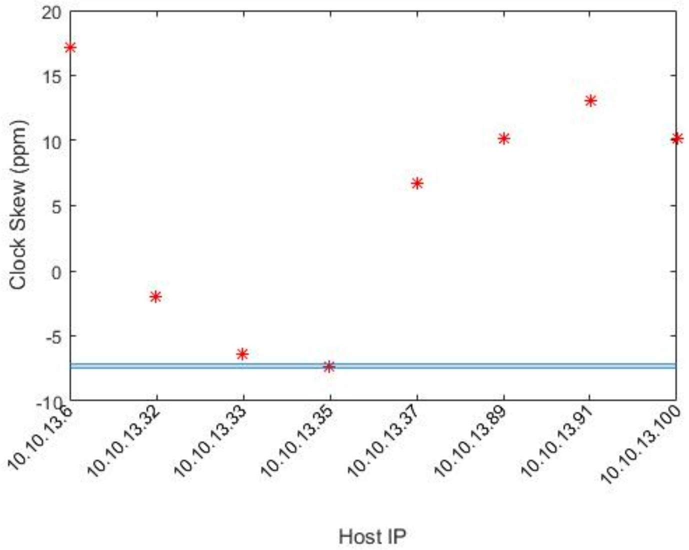


Figure 21. Confidence Interval of 10.10.13.35 Compared to the Mean Value of All Clock Skews Calculated

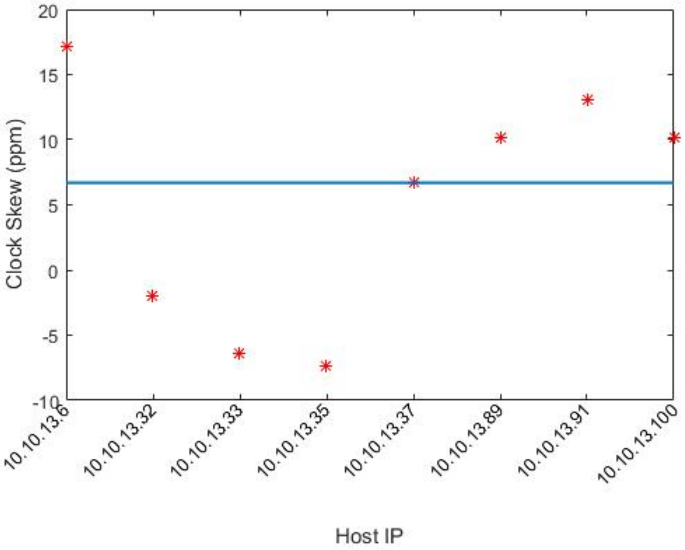


Figure 22. Confidence Interval of 10.10.13.37 Compared to the Mean Value of All Clock Skews Calculated

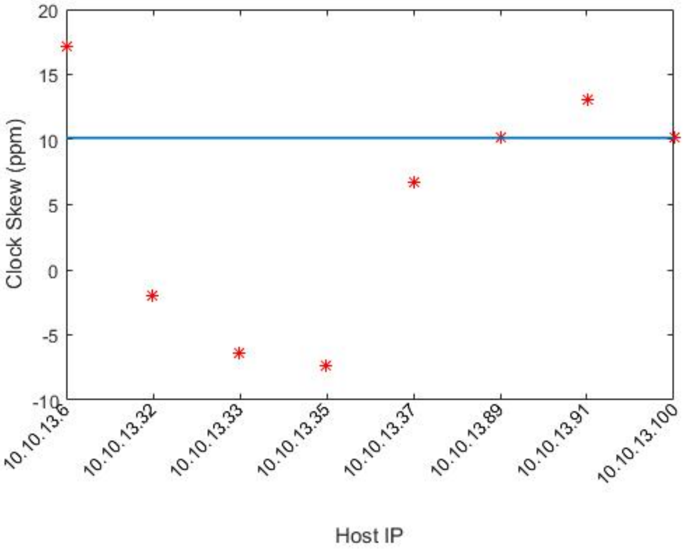


Figure 23. Confidence Interval of 10.10.13.89 Compared to the Mean Value of All Clock Skews Calculated

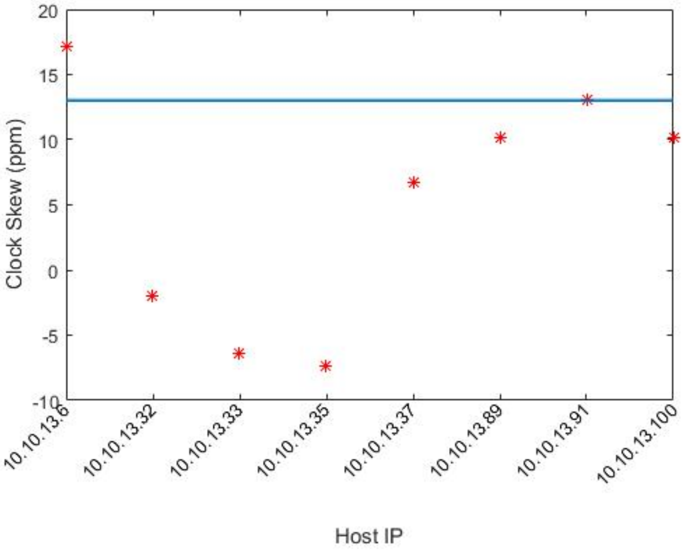


Figure 24. Confidence Interval of 10.10.13.91 Compared to the Mean Value of All Clock Skews Calculated

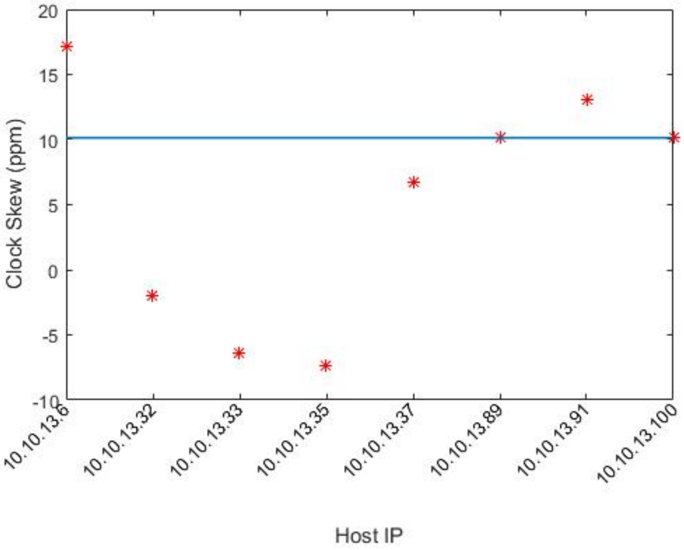


Figure 25. Confidence Interval of 10.10.13.100 Compared to the Mean Value of All Clock Skews Calculated

E. VALIDATING DETECTION SCHEME WITH A SECOND DUAL-HOMED HOST

The results from this testing were validated by moving the dual-homed connection to another device and repeating the proposed detection scheme. The USB 2.0 Gigabit LAN adapter was removed from the host using the IP addresses 10.10.13.89 and 10.10.13.100 to the host that was previously using the IP address 10.10.13.6. This device was now the dual-homed device and was also using the IP address of 10.10.13.89. After generating traffic as in the previous experiment and calculating the clock skews, we determined that the dual-homed connection could still be detected. As shown in Figure 26, the upper bound solutions to (8) for the hosts 10.10.13.89 and 10.10.13.100 are no longer parallel. Instead, the parallel solution has shifted to 10.10.13.6 and 10.10.13.89. This supports the change in network configuration.

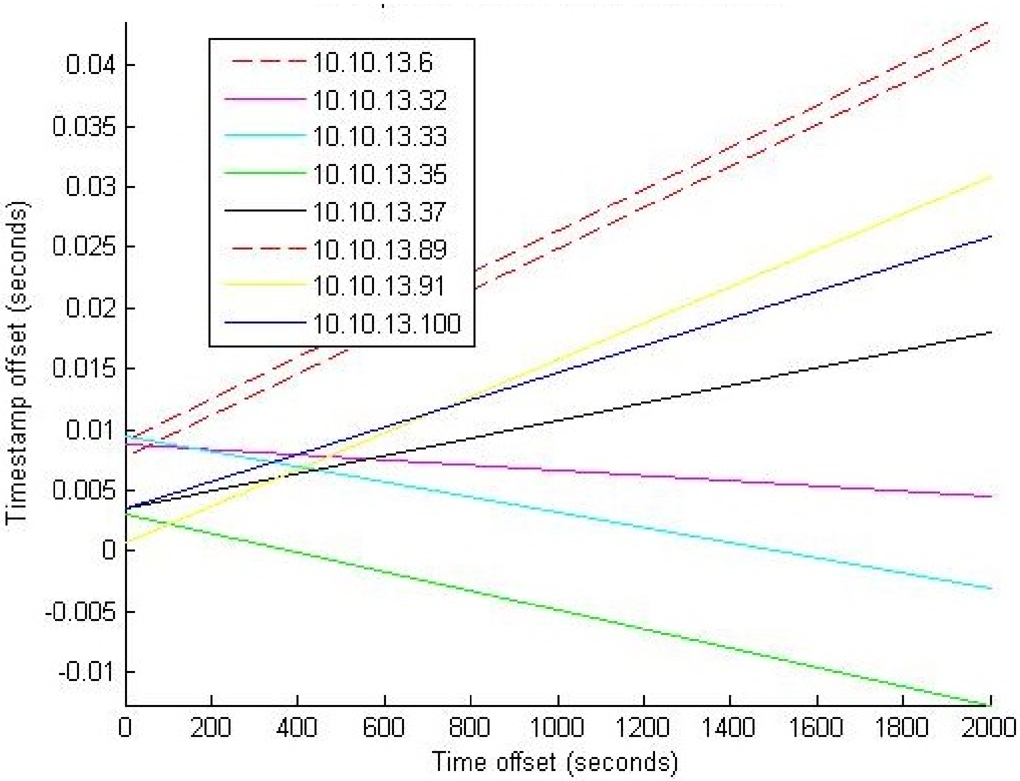


Figure 26. Upper Bound Solution of All Hosts over a Single Trial after the Second Connection was Shifted to the Host with IP address 10.10.13.6

When the confidence intervals of the mean clock skews were compared after 30 trials, the detection scheme was correctly able to identify the dual-homed host as 10.10.13.6 and 10.10.13.89. In Figure 27 the confidence interval for 10.10.13.6 is shown in relation to the clock skew of hosts on the network. This confidence interval contains the clock skews for 10.10.13.6 and 10.10.13.89. This same outcome is shown in Figure 28 where the confidence interval for 10.10.13.89 contains the clock skews for 10.10.13.6 and 10.10.13.89. These results confirm the change in network configuration.

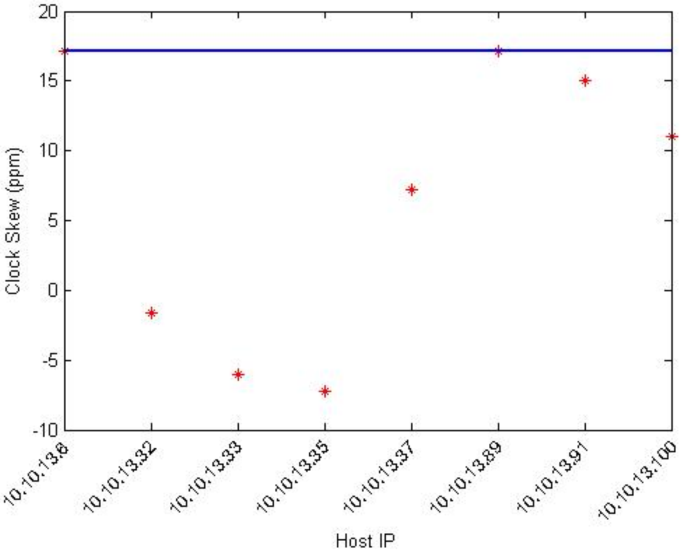


Figure 27. Confidence Interval of 10.10.13.6 After Shifting the Dual Connection Compared to the Mean Value of All Clock Skews Calculated

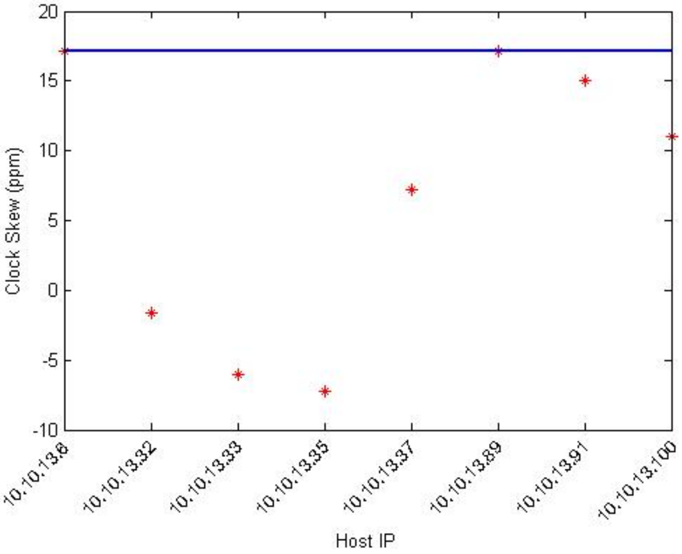


Figure 28. Confidence Interval of 10.10.13.89 After Shifting the Dual Connection Compared to the Mean Value of All Clock Skews Calculated

F. VALIDATING DETECTION SCHEME WITH A MULTI-HOMED HOST

The final validation of the proposed scheme was to add a host with three interfaces to the network and attempt its detection. A Raspberry Pi was connected to the network using its standard built in Ethernet connection as well as with two USB to Ethernet adapters. These interfaces were assigned with the IP addresses of 10.10.13.89, 10.10.13.91, and 10.10.13.100. As in the previous sections, the clock skew for all hosts on the network were calculated, and the proposed scheme was used to correlate any possible multi-home connections. As seen in Figure 29, there are now three parallel lines for the solutions to (8), suggesting that these IP addresses are from the multi-homed host.

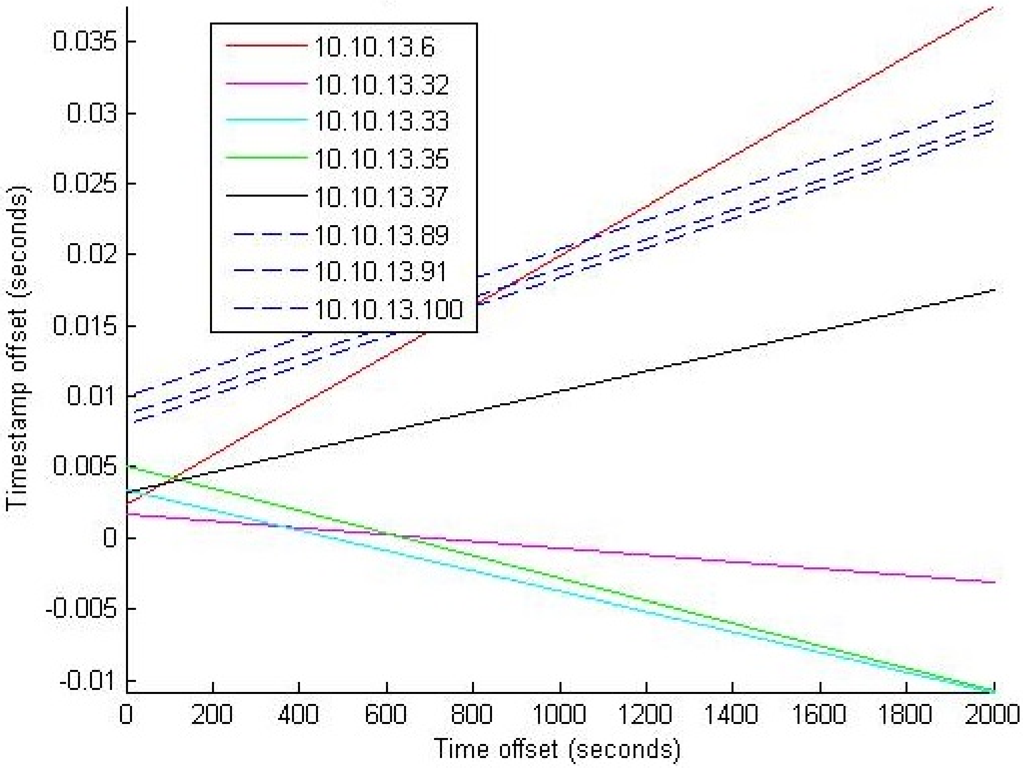


Figure 29. Upper Bound Solution of All Hosts over a Single Trial after the Three Connections Were Made to the Network from One Host

This is confirmed when their mean values are compared to each other's confidence intervals as was done in previous sections. In Figure 30 the confidence interval for 10.10.13.89 is shown to contain the value of the clock skews for 10.10.13.89, 10.10.13.91, and 10.10.13.100. This result is repeated for the confidence interval of 10.10.13.91 in Figure 31 and the confidence interval for 10.10.13.100 in Figure 32. These results confirm the change in network configuration where all three IP addresses are originating from the same device.

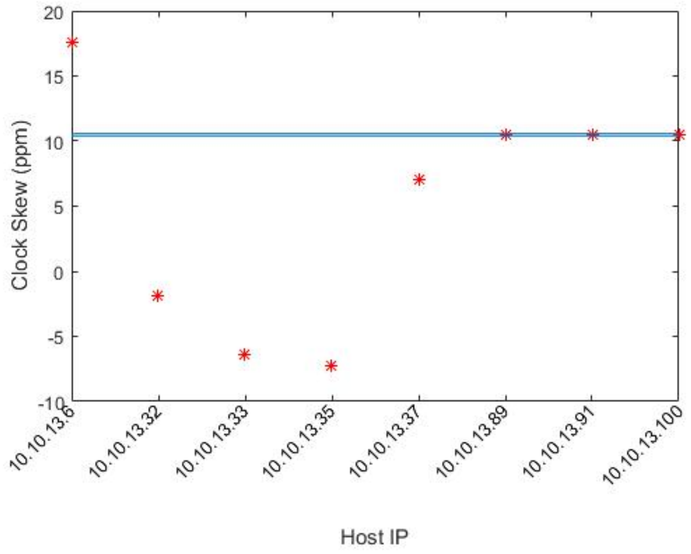


Figure 30. Confidence Interval of 10.10.13.89 When Three Connections Are Made to the Network from One Host

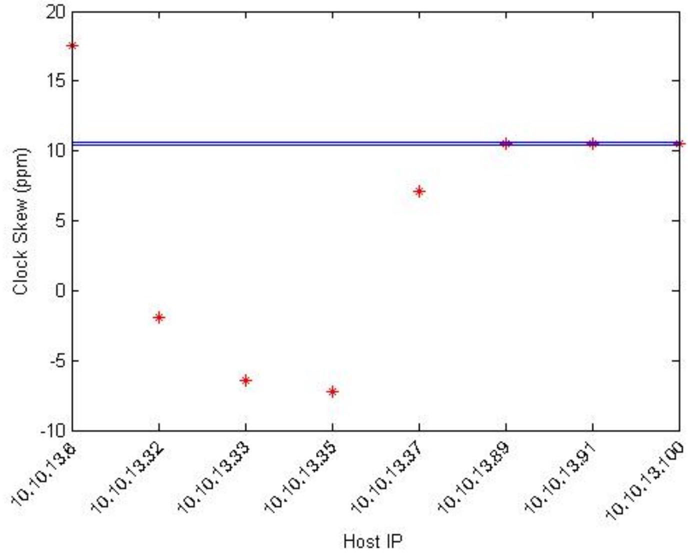


Figure 31. Confidence Interval of 10.10.13.91 When Three Connections Are Made to the Network from One Host

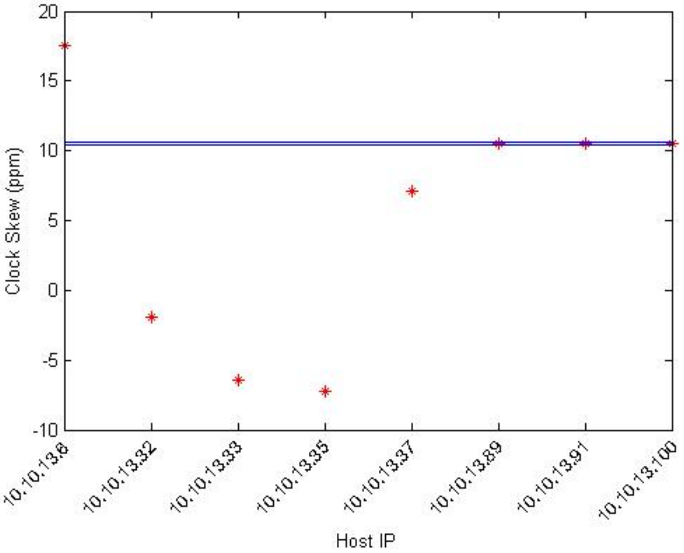


Figure 32. Confidence Interval of 10.10.13.100 When Three Connections Are Made to the Network from One Host

The testing and analysis presented in this chapter demonstrated that clock skew information can be used to identify traffic from different IP addresses that represent the same, multi-homed host. This testing was successfully validated by shifting the multi-homed connection between devices and executing the same methods of detection. Finally, it was shown that the proposed scheme can be used to detect a device using three separate interfaces to connect to the network.

V. CONCLUSION

The idea of using clock skews to remotely identify a device was presented in [5] and further tested in [19]. Continuing this work, we determined that the clock skew of a device can be detected independently of the interface used by that device to connect to the network [6]. This idea was used in previous work for enumeration behind a NAT [7] and was explored in this thesis as a means of detecting a multi-homed host using multiple interfaces.

The motivation for this work was to improve the security of a network and the integrity of its firewall. Since a multi-homed host can be used to bypass a network's firewall and connect directly to the Internet, it is important to be able to detect the presence of such devices. A scheme to use the clock skew of a device as an identifier that is independent of the interface the device used to connect to the network was developed and tested in this work. Since the clock skew of a host stays relatively constant over time [5] and is independent of the interface used [6], it was proposed that this can be used to correlate traffic that appears to be coming from different source IP addresses as traffic from the same host.

A. SIGNIFICANT RESULTS

The proposed detection scheme used network traffic and system clock data in order to identify possible multi-homed hosts on a network. The concept of using clock skew as a unique identifier for a host has been suggested and tested in literature, but this idea has not been utilized in attempting to detect a host on a network using multiple interfaces. These concepts and methods were used to create a model to detect a multi-homed host from a designated fingerprinter. This information can then be used by the controller in a SDN to create new flow rules and isolate a possible multi-homed host for further investigation and to mitigate security risks.

The ability for a designated host to act as a fingerprinter and determine the clock skews of each host on its subnet based on information from its own internal clock and TCP timestamp information was demonstrated in this research. Based on this information, it was shown using analyses of the confidence intervals of a device's clock skew compared to the calculated mean clock skew of all other devices on the network that the traffic from IP addresses that originated from the same host can be correlated to one another.

The detection scheme was then repeated after shifting the dual-homed connection to another device and successfully identifying that host as dual-homed. Finally, it was shown that it was possible to use this scheme to detect a device on the network using three distinct interfaces.

B. RECOMMENDATIONS AND FUTURE WORK

The concept of using clock skews to identify traffic from multiple IP addresses that originated from the same host was presented in Chapter III and tested and validated in Chapter IV. Another means of calculating clock skew is from timestamp data in ICMP packets [5], [22]. This was not tested in this thesis and is another possible means of detection that can be further explored for validation of these results or to improve the granularity of detection.

The proposed scheme was implemented with the fingerprinter that was on the same subnet as all other hosts. What was not shown in this research was the implementation of this process from a panoptic or comprehensive viewpoint such as an SDN controller. It was not demonstrated that the switches used in this network were capable of forwarding OpenFlow packets with TCP header information from the data plane where the hosts and fingerprinter reside to the control plane. A future effort could focus on implementing this scheme in the control plane and designating the controller as the fingerprinter for the entire network. This will provide a means of monitoring for and reacting to the existence of a multi-homed host from a centralized location.

The proposed scheme was tested using seven Raspberry Pis, with one being multi-homed. The next step is to increase the number of hosts on the network for a larger sample size. While increasing the sample size, variety in the types of host used can be introduced. Since all the hosts used in this thesis were of the same type, there was no variation in operating system, motherboard, or network driver. Introducing variety in devices on the network will further validate the proposed scheme.

In this thesis, testing was done using one SDN test bed with the assumption that the fingerprinter could see all traffic on the network. The next step is to detect a multi-homed host that is connected to multiple networks that are separated by a firewall. Detecting the presence of this device would achieve the end goal of identifying a device on a SDN that presents a threat through its ability to bypass the network's security.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX. MATLAB CODE FOR CALCULATING CLOCK SKEW

```
%% Load the data
Data=xlsread(‘test_21April.xls’);
hosts=[4, 5, 6, 8, 9, 43, 89, 100];
%IP addresses observed, 10.10.13.*
L=length(hosts);

%% Calculate the Clock Skew and plot the data
for n=1:L
[row,~]=find(Data==hosts(n));
%Extract data for a given IP address
Hosts=Data(row,:);
%Create a matrix for that data
for k=1:length(Hosts)
x(k) = Hosts(k,3) -Hosts(1,3);
%calculate the time offset
v(k) = Hosts(k,2) -Hosts(1,2);
%calculate the timestamp offset
end
b=ones(length(x),1);
a=[x’ b];
f= [sum(x)/length(x) 1];
I = linprog(f, -a, -v);
%solving the linear programing solution
%for Hz
for k=1:length(Hosts)
w(k) = v(k)/round(I(1));
%adjusting v based on Hz
%the difference between observed and
%actual time
y(k) = w(k) - x(k);

end
z=linprog(f, -a, -y);
%linear programming solution for which
%provides the slope of O, which is the
%clock skew
Z(n)=z(1);
figure
hold on
plot(x,y,’r.’)
%plotting the upper bound limit of O
h=refline(z(1),z(2));
get(h, ‘linewidth’);
```

```
set(h, ‘linewidth’, 2.5);
title([‘Clock Skew for host 10.10.13.’ num2str(hosts(n)) ‘
‘])
xlabel(‘Time offset (seconds)’)
ylabel(‘Timestamp offset (seconds)’)
clear Hosts x v w y b a f I z
end

format long
fprintf(‘ Host Clock Skew \n’)
fprintf(‘%10.6f %15.6f\n’ ,[hosts’ Z’]’)
```

LIST OF REFERENCES

[1] D. Kreutz, F. M. V. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, “Software-defined networking: A comprehensive survey,” *Proceedings of the IEEE*, vol. 103, pp. 14–76, 2015.

[2] E. Byres, “Dual homed machines are the juiciest targets,” *Tofino Security,* 2010. https://www.tofinosecurity.com/blog/dual-homed-machines-are-juiciest-targets

[3] H. Bigdoli, *Handbook of Information: Security, Threats, Vulnerabilities, Prevention, Detection and Management.* Hoboken, NJ: John Wiley and Sons, Inc., 2006.

[4] T. J. Klevinsky, S. Laliberte, and A. Gupta, *Hack I.T.: Security through Penetration Testing.* Boston, MA: Pearson Education, Inc., 2002.

[5] T. Kohno, A. Broido, and K. C. Claffy, “Remote physical device fingerprinting,” *IEEE Transactions on Dependable and Secure Computing*, vol. 2, pp. 93–108, 2005.

[6] L. Polcak, J. Jirasek, and P. Matousek, “Comment on Remote Physical Device Fingerprinting,” *IEEE Transactions on Dependable and Secure Computing*, vol. 11, pp. 494–496, 2014.

[7] G. Wicherski, F. Weingarten, and U. Meyer, “IP agnostic real-time traffic filtering and host identification using TCP timestamps,” in *IEEE 38th Conference on Local Computer Networks*, 2013, pp. 647–654

[8] W. Jianping V. M. Vokkarane, R. Jothi, Xiangtong Qi, B. Raghavachari, and J. P. Jue, “Dual-homing protection in IP-over-WDM networks,” *Journal of Lightwave Technology,* vol. 23, pp. 3111–3124, 2005.

[9] S. Sezer, S. Scott-Hayward, P. K. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller and N. Rao, “Are we ready for SDN? Implementation challenges for software-defined networks,” *IEEE Communications,* vol. 51, pp. 36–43, 2013.

[10] K. Bakshi, “Considerations for software defined networking (SDN): Approaches and use cases,” in *IEEE Aerospace Conference,* 2013, pp. 1–9.

[11] J. L. Johnson, “Software Defined Network Monitoring Scheme Using Spectral Graph Theory and Phantom Nodes,” M.S. thesis, Electrical and Computer Engineering, Monterey, CA: Naval Postgraduate School, 2014.

[12] S. Zander and S. J. Murdoch, “An improved clockskew measurement technique for revealing hidden services.” In *USENIX Security Symposium,* 2008, pp. 211–226.

[13] F. Lanze, A. Panchenko, B. Braatz, and A. Zinnen, “Clock skew based remote device fingerprinting demystified,” in *IEEE Global Communications Conference*, 2012, pp. 813–819.

[14] D. L. Mills, “Network Time Protocol (Version 3): Specification, Implementation, and Analysis,” RFC 1305, 1992.

[15] J. Postel, “Transmission Control Protocol,” STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, http://www.rfc-editor.org/info/rfc793.

[16] Jacobson, V., Braden, R., and D. Borman, “TCP Extensions for High Performance,” RFC 1323, DOI 10.17487/RFC1323, May 1992, http://www.rfc-editor.org/info/rfc1323.

[17] J. Le Boudec, *Performance Evaluation of Computer and Communication Systems,* EPFL Press, Lausanne, Switzerland, 2010.

[18] M. Tummala and C. Therrien, *Probability and Random Processes for Electrical and Computer Engineers.* Boca Raton, FL: CRC Press, 2012.

[19] H. Kikuchi, Y. Tominaga, and Y. Tanaka, “Remote host fingerprinting based on clock skew,” in *International Symposium on Communications and Information Technologies*, 2008, pp. 225–227.

[20] C. W. Therrien, *Discrete Random Signals and Statistical Signal Processing.* New York: Prentice Hall, 1992.

[21] T. C. Parker, “Spectral graph theory analysis of software-defined networks to improve performance and security,” Ph.D. dissertation, Electrical and Computer Engineering, Monterey, CA: Naval Postgraduate School, 2015.

[22] S. Sharma, H. Saran, and S. Bansal, “An empirical study of clock skew behavior in modern mobile and hand-held devices,” in *Third International Conference on Communication Systems and Networks*, 2011, pp. 1–4.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California